Special Issue Proposal for the Information and Software Technology journal on "Agile, Lightweight and Lean Practices for Designing Software Architectures"

Journal Editor in Chief

Prof. Günther Ruhe

Guest Editors

Hector Duran-Limon, University of Guadalajara, Mexico
Claude Y. Laporte, École de technologie supérieure, Canada
Orlando Karam, Amazon AWS, USA
Manuel Mora, Autonomous University of Aguascalientes, Mexico
Alok Mishra, Molde University College-Specialized University in Logistics, Norway and Atilim University, Turkey

Theme

Software Architecture artifacts are considered as the first-class type in the classic and plan-driven software development methodologies (Clemens et al., 2011). A first-class artifact implies that the artifact is considered mandatory to be elaborated in a high-quality system development methodology, because its omission or partial, or low-quality elaboration leads to a flawed software design and lately to a wrong software product. A vast body of knowledge on Software Architecture (SA) design and evaluation methods exists in the SA literature from a classic and plan-driven approach (Hofmeister et al., 2007; Shahin et al., 2014; Hohpe et al., 2016). In contrast, in the Agile software development approach (Hoda et al., 2018), Software Architecture design and evaluation methods, as well as Software Architecture artifacts, are still under academic and practitioner debate (Kruchten, 2010; Abrahamsson et al., 2010; Yang et al., 2016).

Agile approaches have also succeeded in having a more predictable product delivery, schedule, and costs. Agile processes are very dynamic and flexible and eliminate much of the bureaucratic documentation procedures of rigor-oriented methodologies. Therefore, Agile development has gained large acceptance in Industry as a methodology to increase adaptability to changes as opposed to the rigidity of rigor-oriented methodologies (Hoda et al., 2018). However, whereas a well-documented software architecture design activity is mandatory in plan-driven methodologies (Reyes-Delgado et al., 2016), this is not promoted by agile practices (Kruchten, 2010;

Abrahamsson et al., 2010). This omission is considered a weakness of current agile approaches, given the introduction of the *technical debt problem* (Behutiye et al., 2017) and because it is widely recognized that software architecture design practices help to avoid costly errors to be corrected in the subsequent software process development phases (Boehm, 2002).

To alleviate software architecture design omissions in agile practices, lightweight and lean approaches have been suggested. Lightweight and lean are necessary but not sufficient attributes for agile practices (Qumer and Henderson-Sellers, 2008; Conboy 2009), and thus lightweight and lean practices can be considered in the middle point between agile and plan-driven methods (Wang et al., 2012). The new ISO/IEC 29110 standard series for Very Small Entities (ISO/IEC, 2015) and the Disciplined Agile Framework (Ambler and Lines, 2016) are interesting examples of lightweight and lean approaches, respectively. Furthermore, agile practices have been challenged to be used in large-scale projects, and the inclusion of lightweight and lean approaches are promoted (Gill et al., 2018; Dingsøyr et al., 2019).

Hence, it remains an important challenge to incorporate software architecture procedures into current agile practices while maintaining similar levels of flexibility (Yang et al., 2016).

This special issue aims to highlight the importance and utilization enablers and inhibitors of software architecture design methods in the agile, lightweight, and lean approaches, and its impact on the six-pointed star model (cost, time, quality, scope, risk, and resource) (Akbar et al., 2018).

Thus, this special issue calls for unpublished contributions of high scientific quality, as well as practical and industrial experience reports addressing the agile, lightweight, or lean software development, in any of the following areas, that include, but not limited to:

- Fundamental studies on the tenet of software architecture practices.
- Conceptual reviews of software architecture practices.
- Comparative studies against software architecture practices in the plan-driven approach.
- Conceptual frameworks on software architecture practices.
- Empirical survey with practitioners of software architecture practices.
- Experimental research comparing software architecture practices.
- Predictive and Explanatory simulation-based studies on software architecture practices.
- Industrial and academic case studies of software architecture practices.
- New methods of software architecture practices.
- Data Science studies on data collected from projects using software architecture practices.
- Utilization of software architecture practices combined with innovative practices like DevOps, microservices, SaaS, among others.

- Studies on the profiles of agile Software Architects.
- Agile software architecture practices in the cloud.

References

Abrahamsson, P., Babar, M. A., & Kruchten, P. (2010). Agility and architecture: Can they coexist?. IEEE Software, 27(2), 16-22.

Akbar, M.A., Sang, J., Khan, A.A., Amin, F.E., Hussain, S., Sohail, M.K., Xiang, H., & Cai, B. (2018). Statistical analysis of the effects of heavyweight and lightweight methodologies on the six-pointed star model. IEEE Access, 6, 8066-8079.

Ambler, S. W., & Lines, M. (2016). The disciplined agile process decision framework. In International Conference on Software Quality (pp. 3-14). Springer, Cham.

Behutiye, W. N., Rodríguez, P., Oivo, M., & Tosun, A. (2017). Analyzing the concept of technical debt in the context of agile software development: A systematic literature review. Information and Software Technology, 82, 139-158.

Boehm, B. (2002). Get ready for agile methods, with care. Computer, 35(1), 64-69.

Clements, P., Bachmann, F., Bass, L., Garlan, D., Ivers, J., Little, R., Merson, P., Nord, R., & Stafford, J. (2011). Documenting Software Architecture: Views and Beyond, Boston, MA, USA: Addison Wesley, pp. 01-517

Conboy, K. (2009). Agility from first principles: Reconstructing the concept of agility in information systems development. Information systems research, 20(3), 329-354.

Dingsøyr, T., Falessi, D., & Power, K. (2019). Agile development at scale: the next frontier. IEEE Software, 36(2), 30-38.

Gill, A. Q., Henderson-Sellers, B., & Niazi, M. (2018). Scaling for agility: A reference model for hybrid traditional-agile software development methodologies. Information Systems Frontiers, 20(2), 315-341.

Hoda, R., Salleh, N., & Grundy, J. (2018). The rise and evolution of agile software development. IEEE Software, 35(5), 58-63.

Hofmeister, C., Kruchten, F., Nord, R., Obbink, H., Ran, A. & America, P. (2007). A general model of Software Architecture design derived from five industrial approaches. Journal of System and Software, 80(1), 106-126.

Hohpe, G., Ozkaya, I., Zdun, U., & Zimmermann, O. (2016). The software architect's role in the digital age. IEEE Software, 33(6), 30-39.

Kruchten, P. (2010). Software architecture and agile software development: a clash of two cultures?. In Proceedings of the 32nd ACM/IEEE International Conference on Software Engineering-Volume 2 (pp. 497-498).

ISO/IEC (2015). ISO/IEC 29110-2-1:2015: Software engineering - Lifecycle profiles for Very SmallEntities (VSEs) - Part 2-1: Framework and taxonomy. Standard, International Organization for Standardization, Geneva.

Qumer, A., & Henderson-Sellers, B. (2008). An evaluation of the degree of agility in six agile methods and its applicability for method engineering. Information and software technology, 50(4), 280-295.

Reyes-Delgado, P. Y., Mora, M., Duran-Limon, H. A., Rodríguez-Martínez, L. C., O'Connor, R. V., & Mendoza-Gonzalez, R. (2016). The strengths and weaknesses of software architecture design in the RUP, MSF, MBASE and RUP-SOA methodologies: A conceptual review. Computer Standards & Interfaces, 47, 24-41.

Shahin, M., Liang, P., & Babar, M. A. (2014). A systematic review of software architecture visualization techniques. Journal of Systems and Software, 94, 161-185.

Yang, C., Liang, P., & Avgeriou, P. (2016). A systematic mapping study on the combination of software architecture and agile development. Journal of Systems and Software, 111, 157-184.

Wang, X., Conboy, K., & Cawley, O. (2012). "Leagile" software development: An experience report analysis of the application of lean approaches in agile software development. Journal of Systems and Software, 85(6), 1287-1299.

Important dates

- First submission deadline: 30 November 2021
- First editorial notification: 30 March 2022
- Revised manuscript submission deadline: 31 May 2022
- Final editorial notification: 30 June 2022
- Camera-ready version submission deadline: 31 July 2022

Submission Guidelines:

The call for this special issue is an open call. We invite authors interested in this relevant theme to submit original and unpublished research of high quality in concordance with the quality standards of the …

Guest Editors' Summarized Curriculum Vitae

Hector Duran-Limon is a Full Professor at the Information Systems Department, University of Guadalajara. He completed a Ph.D. at Lancaster University in 2002. Following this, he was a post-doctoral researcher until December 2003. He obtained an IBM Faculty award in 2008. His research interests include Grid Computing, Adaptive Middleware, and Mobile Computing. He is also interested in software architectures and component-based development. In 2006, He was invited to create a Ph.D. program in Information Technologies for the University of Guadalajara, becoming a member of the Academic Council. Contact him at the Information Systems Department, University of Guadalajara, Mexico.

Claude Y. Laporte is a Professor at the École de technologie supérieure (ETS) an engineering school of the Université du Québec network of institutions. He teaches graduate and undergraduate software engineering courses. He did his undergraduate studies at the Collège Militaire Royal de St-Jean (Canadian Military College). He received his Master in Applied Sciences from the Electrical and Computer Engineering department of École Polytechnique de Montréal in 1986. In 1980, he received his Master in Physics (computerized instrumentation) from the Université de Montréal. He completed a Ph.D., under the supervision of Dr. J. Tisseau, Director of the National Engineering School of Brest (ENIB) of the Université de Bretagne Occidentale (UBO). He received a Doctor Honoris Causa from the Faculty of Engineering and Architecture of the Universidad de San Martin de Porres (Lima, Peru). In 1989, he instigated the development of a software engineering center modeled on the Software Engineering Institute. This led to the establishment of the Applied Software Engineering Center (ASEC) of the Computer Research Institute of Montreal (CRIM). Between 1978 and 1991, he was a professor at the Royal Military College (Saint-Jean), a Military university of the Department of National Defense. He was tasked in 1988, as a project manager, to lead the development of a graduate program in software engineering for the Department of National Defense. From 1992 to 1999, he worked at Oerlikon Aerospace (now Rheinmetall Defense) as a senior analyst responsible to coordinate the development and deployment of software and systems engineering processes and management processes. He participates, as an Affiliate of the Software Engineering Institute, in the project Improving Processes for Small Settings. He co-founded and chaired the Montréal Software Process Improvement Network (SPIN). SPINs are sponsored by the Software Engineering Institute. His interests include the development and deployment of software and systems engineering processes, process assessment, software quality, and the management of technological changes. He is a life member of the IEEE Computer Society (IEEE), the International Council on Systems Engineering (INCOSE), and the Project Management Institute (PMI). He has been an invited professor at the French University of Brest (Université de Bretagne Occidentale) and the Chiang Mai University of Thailand. Since 2005, he is the Project Editor of an ISO Working Group, Working Group 24, of ISO/SC7/JTC1. The working group has been mandated to develop the ISO/IEC 29110 series of Systems engineering and Software engineering standards for Very Small Entities (organizations with up to 25 people). He also represented the Standards Council of Canada (SCC) at Working Group 20 of JTC1/ISO/SC7. Working Group 20 has developed a standard for the international certification of software engineering professionals. He has been the ISO co-project editor of the Guide to the Software Engineering Body of Knowledge (SWEBOK Guide).

Orlando Karam is a Senior Technical Curriculum Developer at Amazon Web Services. Orlando got his PhD in Computer Science from Tulane University in 2001, and has taught in several universities in the USA. Orlando has also been a software developer at Microsoft and Fabric.com. Orlando has published several papers and a Software Engineering textbook.

Manuel Mora is a full-time Professor in the Information Systems Department at the Autonomous University of Aguascalientes (UAA), Mexico. Dr. Mora holds an M.Sc. in Computer Sciences (Artificial Intelligence area, 1989) from Monterrey Tech (ITESM), and an Eng.D. in Engineering (Systems Engineering area, 2003) from the National Autonomous University of Mexico (UNAM). He has published over 90 research papers in international top conferences, research books, and journals such as IEEE-TSMC, European Journal of Operational Research, Int. Journal of Information Management, Engineering Management, Int. J. of Information Technology and Decision Making, Information Technology for Development, Int. J. in Software Engineering and Knowledge Engineering, and Computer Standards & Interface. Dr. Mora is a senior member of ACM (since 2008), a Mexican SNI at Level II, and serves in the ERB of several international journals indexed by Emergent Source Citation Index focused on decision-making support systems (DMSS) and IT services systems.

Alok Mishra is Professor in Informatics and Digitalization at Molde University College (A Specialized University in Logistics) Norway. He is also associated as Professor in Software Engineering at Atilim University, Turkey. His areas of research interests are software engineering, information system, information technology, and artificial intelligence. Prof. Mishra is an editorial board member of many reputed journals including Computer Standards and Interfaces (Elsevier), Journal of Universal Computer Science, Computing & Informatics, Data Technologies and Applications Journal. He is actively involved in editing special issues of reputed journals in his areas of research interest. Prof. Mishra had also extensive experience in online education related to Computing and Management disciplines. In teaching, he has received excellence in online education award by U21Global Singapore while in research he has been awarded by the Scientific and Research Council of Turkey and Board of Management of University for outstanding publications in Science and Social Science Citation Indexed (Thomson Reuter) journals. He is a recipient of many scholarships, international awards, and research projects.